

е

- алгоритмы**
- ✓ Алгоритм Евклида
 - ✓ Решето Эратосфена
 - ✓ Длинные числа
 - ✓ Последовательность Фибоначчи
 - ✓ Последовательность
КРАСНОУРСКОГО

Целочисленные е алгоритмы

Тема 1. Алгоритм Евклида



Вычисление

НОД

НОД (наибольший общий делитель двух натуральных чисел) – это наибольшее число, на которое оба

исходных числа делятся без остатка.

Перебор.

```
static void Main(string [ ] args)
```

```
{
```

```
    int a = 100;
```

```
    int b = 86;
```

```
    int k = a;
```

```
    while (a % k != 0 || b % k != 0)
```

```
        k--;
```

```
    Console.WriteLine("НОД числа " + a + " и числа " + b + ": " + k);
```

```
    Console.ReadKey(); //Чтобы увидеть результат на экране
```

```
}
```



Алгоритм

Евклида

$$\text{НОД}(a, b) = \text{НОД}(a - b, b)$$

$$= \text{НОД}($$

Заменяем **большее** из двух чисел **разностью** большего и меньшего до тех пор, пока они

не станут равны. Это и есть

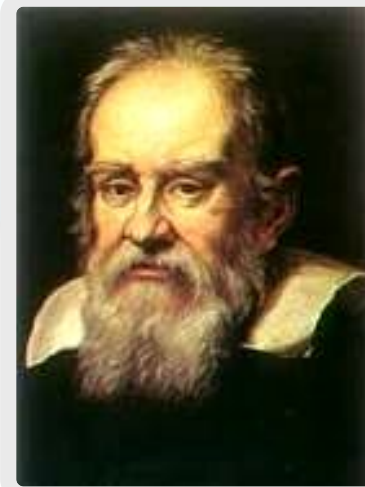
$$\begin{aligned} \text{НОД}(14, 21) &= \text{НОД}(14, 21 - 14) = \text{НОД}(14, 7) \\ &= \text{НОД}(7, 7) = \end{aligned}$$



много шагов при большой

разнице чисел:

$$\text{НОД}(2014, 2) = \text{НОД}(2012, 2) = \dots = 2$$



Евкл

(ок. 325 – 265 гг. до н. э.)

ид



Модифицированный Алгоритм

Евклида

$$\text{НОД}(a, b) = \text{НОД}(a \bmod b, b)$$

$$= \text{НОД}($$

Заменяем большее из двух чисел **остатком от деления** большего на меньшее до тех пор, пока меньшее не станет равно нулю.

Пример

Тогда большее — это НОД.

$$\text{НОД}(14, 21) = \text{НОД}(14, 7) = \text{НОД}(0, 7) =$$

Еще один

$$\text{НОД}(2a, 2b) = 2\text{НОД}(a,$$

$$\text{НОД}(2a, b) = \text{НОД}(a, b) // \text{при нечетном}$$

b

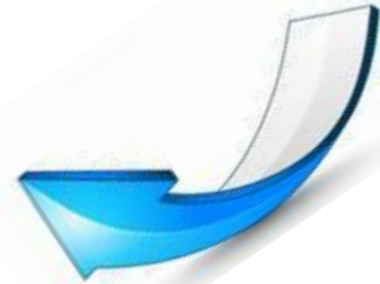


Реализация алгоритма

Евклида

Без рекурсии

```
class Program
{
    static int NOD(int a, int b)
    { while (a * b != 0)
      {
          if (a > b) a = a % b;
          else b = b % a;
      }
      return a + b;
    }
    static void Main(string[] args)
    { Console.WriteLine("a = ");
      int a = Convert.ToInt32(Console.ReadLine());
      Console.WriteLine("b = ");
      int b = Convert.ToInt32(Console.ReadLine());
      Console.WriteLine("Наибольший общий делитель = ");
      Console.WriteLine(NOD(a, b));
      Console.ReadKey();
    }
}
```



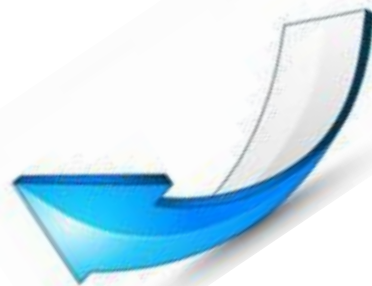
Реализация алгоритма

Евклида

```
class Program
```

```
{  
    static int NOD(int a, int b)  
    {  
        if (a * b == 0)  
            return a + b;  
        if (a > b)  
            return NOD(b, a % b);  
        else  
            return NOD(a, b % a);  
    }  
    static void Main(string[] args)  
    {  
        Console.Write("a = ");  
        int a = Convert.ToInt32(Console.ReadLine());  
        Console.Write("b = ");  
        int b = Convert.ToInt32(Console.ReadLine());  
        Console.Write("Наибольший общий делитель = ");  
        Console.WriteLine(NOD(a, b));  
        Console.ReadKey();  
    }  
}
```

Рекурсивный
способ



Целочисленные алгоритмы

Тема 2

Решето Эратосфена



Поиск простых

Численность числа – это числа, которые делятся только на себя и на 1.

Наибольшее известное (апрель 2014 года):

и содержит 17 425 170

десятичных цифр

Задача. Найти все простые числа в интервале от 1 до заданного N.

решение

```
static void Main(string[] args)
{
    Console.WriteLine("n = ");
    bool isPrime = true;
    int n =
    Convert.ToInt32(Console.ReadLine());
    for (int i = 1; i <= n; i++)
    {
        for (int k = 2; k < i; k++)
        {
            isPrime = true;
```

```
        if (i % k == 0)
        {
            isPrime = false;
            break;
        }
    }
    // если переменная «true»
    if (isPrime)
        Console.WriteLine(i + " ");
    Console.ReadKey();
}
```



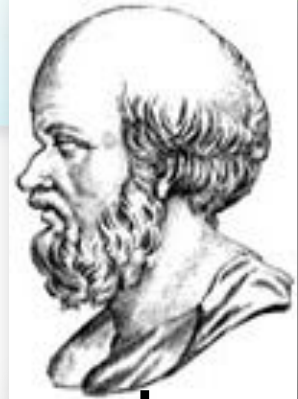
Решето

Эратосфена

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Алгоритм:

- 1) начать с $k = 2$;
 - 2) «выколоть» все числа через k , начиная с $2 \cdot k$;
 - 3) перейти к следующему «невыколотому» k ;
 - 4) и $k \cdot k \leq N$, то перейти к шагу 2;
 - 5) высокая скорость, количество операций печатать все числа, оставшиеся «выколотыми».
- $O((N \cdot \log N) \cdot \log \log N)$
- нужно хранить в памяти все числа от 1 до N



Эратосфен

Киренский

(ок. 275-194 до н.э.)



Реализация алгоритма

```
static void Main(string[] args)
{
    Console.WriteLine("Введите n:");
    int n =
    Convert.ToInt32(Console.ReadLine());
    int[] a = new int[n + 1];
    // сначала все числа не выколоты
    for (int i = 1; i <= n; i++)
        a[i] = 1;
    // основной цикл
    for (int k = 2; k * k <= n; k++)
        if (a[k] != 0)
            for (int i = k + k; i <= n; i += k)
                a[i] = 0;
```

```
// выводим оставшиеся числа
for (int i = 1; i <= n; i++)
{ if (a[i] == 1)
    Console.WriteLine("\n" + i);
}
Console.ReadKey();
}
```

Массив **A[N+1]**, где
A[i]=1, если число **i** не
 «ВЫКОЛОТО»,

A[i]=0, если число **i**
 «ВЫКОЛОТО».



Целочисленные е алгоритмы

Тема 3

Длинные числа



Что такое длинные

числа?

Задача. Вычислить (точно)

$$100! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot 99 \cdot 100$$

Проблема:

это число содержит более 100 цифр...

Решение:

хранить цифры в виде массива, по группам (например, 6 цифр в ячейке).

$$100! < \underbrace{100^{100}}_{201 \text{ цифра}}$$

$$201/6 \approx 34$$

ячейки



Хранение длинных

Чисел

$$1234568901734567 = 1234 \cdot 1000000^2 +$$

$$+ 568901 \cdot 1000000^1 + 734567 \cdot 1000000^0$$

Хранить число по группам из 6 цифр – это значит представить его в системе счисления с основанием $d = 1000000$.

Алгоритм

ТМ:

```
{A} = 1;
for ( k = 2; k <= 100; k ++ )
    { A} = {A} * k;
... // вывести {A}
```

{A} – длинное число,
хранящееся как массив

умножение длинного
числа на «короткое»



Вычисление

```
n!  
class Program  
{  
    static int Factorial(int n)  
    {  
        int result = 1;  
        for (int i = 1; i <= n; i++)  
        {  
            result = result * i;  
        }  
        return result;  
    }  
    static void Main(string[] args)  
    {  
        Console.Write("Введите n: ");  
        int n = Convert.ToInt32(Console.ReadLine());  
        Console.Write(n + "! = ");  
        Console.WriteLine(Factorial(n));  
        Console.ReadKey();  
    }  
}
```

15!
=
2004310
016



Целочисленные алгоритмы

Тема 4

Последовательность Фибоначчи





Леонардо Пизанский (Фибоначчи) (1180 - 1240)

Итальянский купец Леонардо из Пизы был одним из самых известных математиков средневековья.



Числа

Фибоначчи

Числа Фибоначчи — элементы числовой последовательности,

в которой каждое последующее число равно

сумме двух предыдущих чисел.
 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,
 89, 144, 233, 377, 610,

987, 1597, 2584, 4181,

| | | | | | | | | | |
|---|---|---|---|---|----|---|---|---|-----|
| 6 | 7 | 6 | 5 | 2 | 10 | 9 | 4 | 6 | ... |
| | 3 | | | 1 | 1 | | | | |
| | | | | | | | | 8 | |
| | | | 5 | | | | | | |

Геометрическое доказательство формулы для суммы квадратов первых n чисел Фибоначчи



Числа

Фибоначчи

Более формально, последовательность чисел

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}, \quad n \geq 2, \quad n \in \mathbb{Z}.$$

соотношением:

Задача 1:

Вывести на экран все числа последовательности Фибоначчи до N-го числа последовательности (использовать динамическое

Задача 2:

программирование)
Вывести на экран N -ое число последовательности Фибоначчи (рекурсией)



Решение задачи

№ 1

```
public static void Fib(double[] a)
{
    double n = a.Length;
    a[0] = 1;
    a[1] = 1;
    Console.WriteLine("1 1");
    for (int i = 2; i <= n; i++)
    {
        a[i] = a[i - 1] + a[i - 2];
        Console.WriteLine(" " + a[i]);
    }
}
```

```
static void Main(string[] args)
{
    Console.WriteLine("Введите нужное  
количество чисел (n): ");
    int n = Convert.ToInt32(Console.  
ReadLine());
    Console.WriteLine(" Первые " + n +  
" чисел последовательности  
Фибоначчи: ");
    double[] a = new double[n];
    Fib(a);
    Console.ReadKey();
}
```



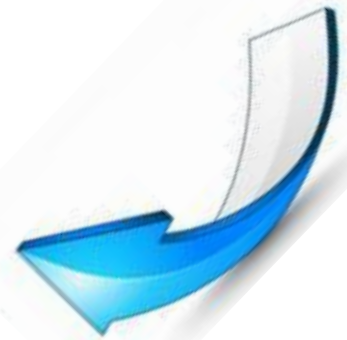
Решение задачи

№2

class Program

```
{
    static private int Fib(int x)
    {
        if (x == 1 || x == 2)
            return 1;
        return Fib(x - 2) + Fib(x - 1);
    }
    static void Main(string[] args)
    {
        Console.Write("Введите номер нужного числа (n):");
        int n = Convert.ToInt32(Console.ReadLine());
        Console.Write(n + "-ое число последовательности Фибоначчи = ");
        Console.WriteLine(Fib(n));
        Console.ReadKey(true);
    }
}
```

Рекурсивный
способ



Целочисленные алгоритмы

Тема 5

Последовательность квадратов



Последовательность

Квадратов

Задача:

Вывести на экран квадраты всех чисел от 0 до N

```
static void Main(string[] args)
{
    Console.Write("Введите n: ");
    long n =
    Convert.ToInt64(Console.ReadLine());
    long t = 1;
    for (long k = 0; k <= n; k++)
    {
        t = t + 2 * k - 1;
        Console.WriteLine(k + "^2 = " + t);
    }
    Console.ReadKey();
}
```

Реализация
алгоритма



The end

